

## LA-UR-17-28235

Approved for public release; distribution is unlimited.

Title: Physics Verification Overview

Author(s): Doebling, Scott William

Intended for: Milestone Review

Issued: 2017-09-12

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Physics Verification Overview

## Physics V&V L2 Milestone Review

**Presented by: Scott Doebling**

September 14, 2017



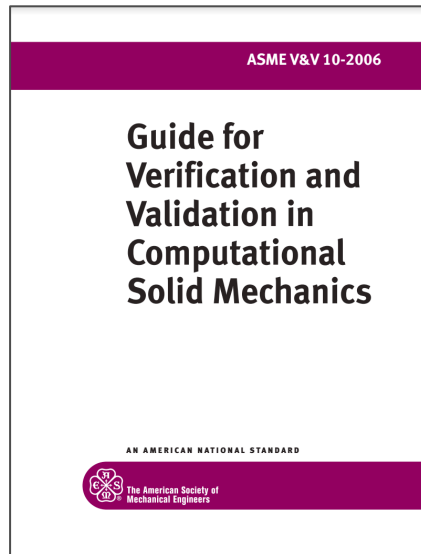
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

# Overview of the Physics Verification Project

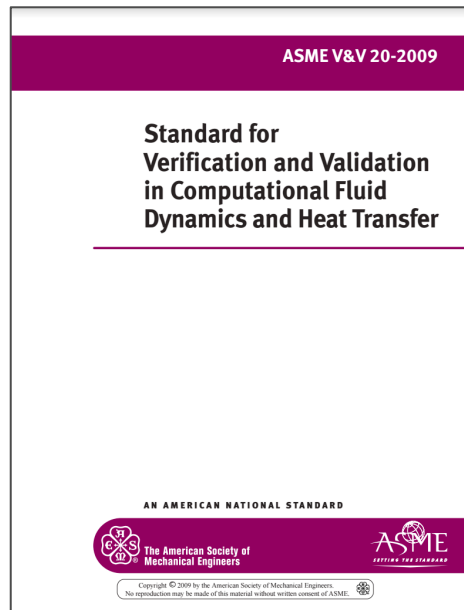
## Mission

1. Establish, through rigorous convergence analysis, that each ASC computational physics code correctly implements a set of physics models and algorithms (code verification)
  2. Evaluate and analyze the uncertainties of code outputs associated with the choice of temporal and spatial discretization (solution or calculation verification),
  3. Develop and maintain the capability to expand and update these analyses on-demand.
- **This mission is accomplished via the development and maintenance of software tools, code test suites, and verification assessments.**
  - **Focus areas for this milestone**
    - Develop software tools to support code verification analysis
    - Document standard definitions of code verification test problems
    - Perform code verification assessments (focusing on error behavior of algorithms)

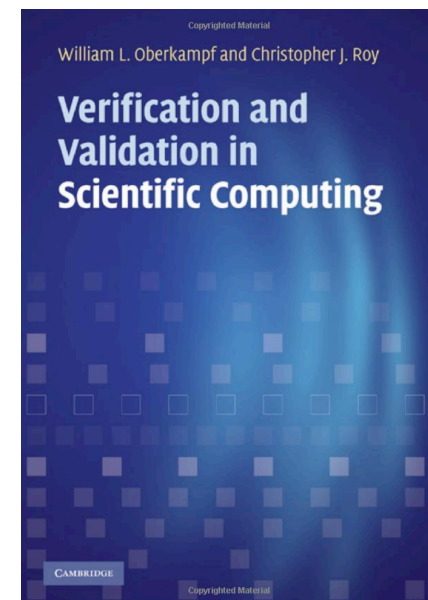
# Published standards provide a foundation for verification methods



ASME V&V 10-2006



ASME V&V 20-2009



Oberkampf & Roy, 2010

# Verification criteria in order of rigor

## 1. Simple tests

- Symmetry, conservation, Galilean invariance

## 2. Code-to-code comparisons

- Typically against a trusted reference code

## 3. Discretization error quantification


- Compute error norm between code output and known solution

## 4. Convergence tests

- Does code converge to known solution under refinement?

## 5. Order of accuracy tests

- Does code converge to known solution at expected theoretical rate?



Less rigorous,  
but can be applied  
to a wide variety of  
test cases

More rigorous,  
but require exact  
solution to use as  
reference

# Order of accuracy tests are the most rigorous

- **Formal order of accuracy**

- Theoretical calculation of truncation error between the discrete equations and the original PDE

- **Observed order of accuracy**

- Rate at which discrete solution approaches exact solution as discretization parameters go to zero
- Estimated using the discretization error observed when systematically refining the computational mesh
- There are other errors embedded in the numerical methods. These will be discussed later.

***Comparing formal and observed orders of accuracy is a key part of the process***

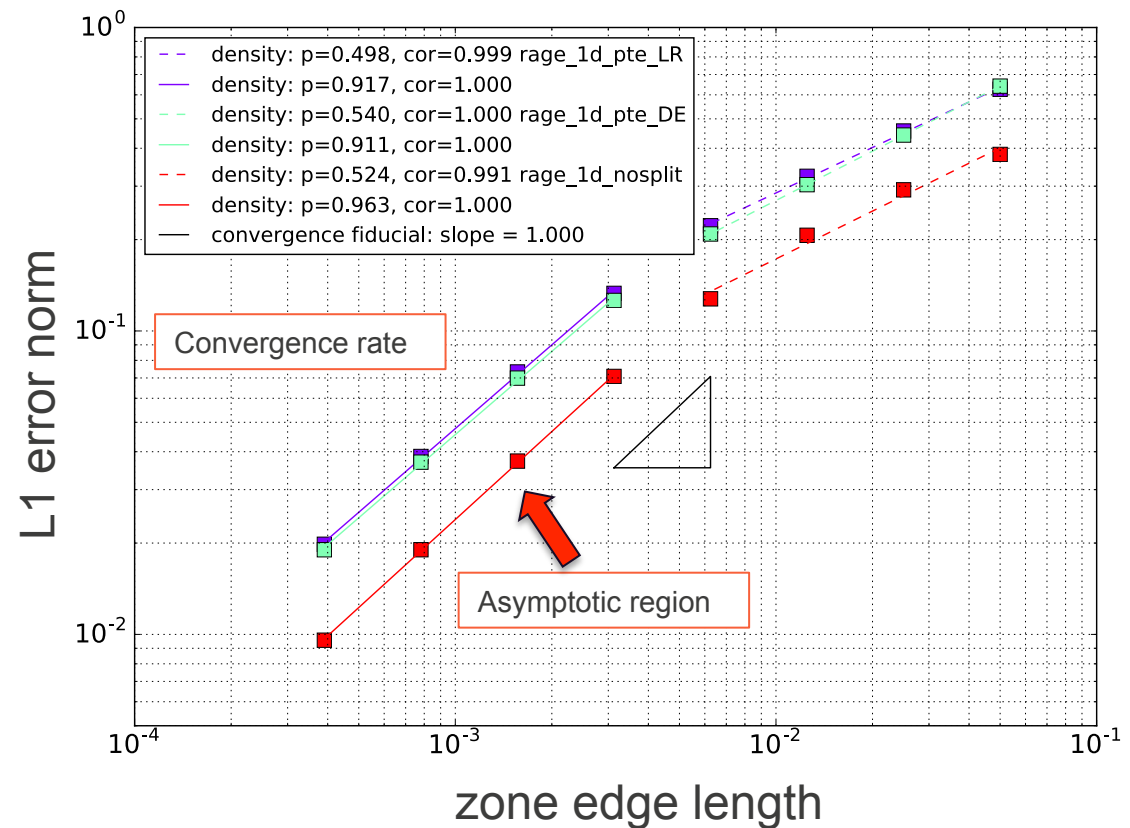
# **Exact solution problems are the most straightforward method for rigorous verification**

- **For rigorous verification (Levels 3-5), known solution to underlying governing equations is required**
- **Exact solution problems are a convenient mechanism for this**
- **Solutions can be analytic (closed-form) or semi-analytic (solution to ODE)**
- **These problems are a good starting point for code verification**
- **But they are limited in dimensionality, physics complexity, etc.**
- **Additional approaches that preserve rigor while increasing complexity will be discussed in final wrap-up**



# Convergence plots provide a visual illustration of verification rigor

- Identification of the asymptotic region is important to understand error behavior of the algorithm as implemented
- Most verification test problems are scale invariant, so don't attribute meaning to the absolute magnitude of the zone size
- Recall verification criteria:
  - Level 3: Quantify error?
  - Level 4: Converges?
  - Level 5: Conv. rate?

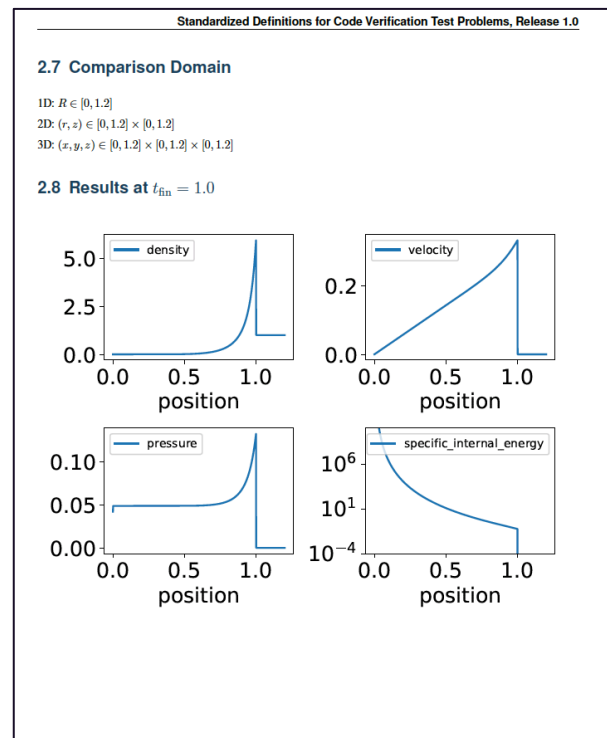
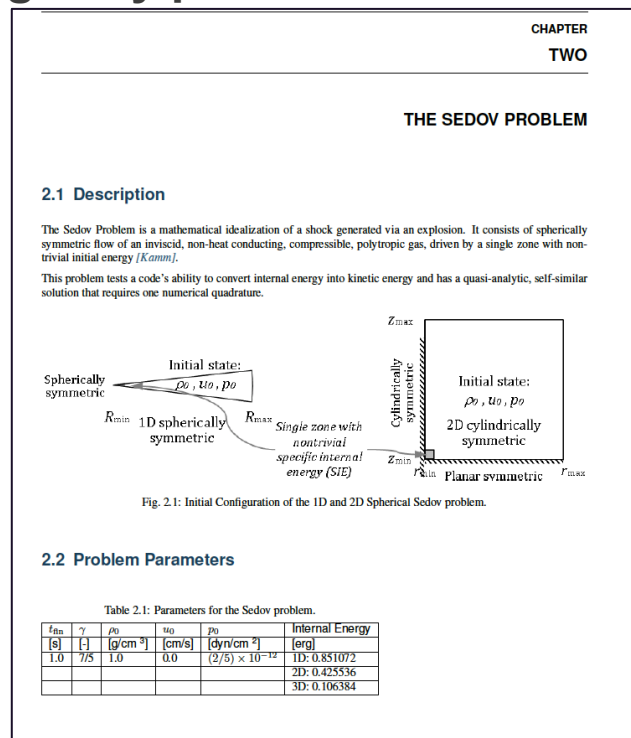


# The Verification Test Suite

- To cover a broad variety of the capabilities in a computational physics code, a *suite* of verification test problems is required
- A Verification Test Suite requires the following components:
  - Standardized definitions for verification test problems
    - We call these the “one-pagers”
    - Intended to define implementation in “any code”
  - Test problem exact solutions
    - Our code ExactPack calculates these
  - Input deck generation & simulation execution
    - This is managed by our code LSPDM
  - Code verification analysis
    - ExactPack provides tools for this
  - Documentation and archiving
    - This is also managed by LSPDM

# We have published a set of standardized verification test problem definitions

- Setup specification for each problems intended to be code-independent
  - This means that often it is not set up optimally for a particular code
- Originally published as LA-UR-14-20418 – new version imminent



**The following two talks will illustrate key aspects of the Verification Test Suite work**

**Software Infrastructure for V&V (Dan Israel)**

**Verification Assessments using the Verification Test Suite (Nate Woods)**